

METADATA IN JPEG2000 FILE FORMAT

5 CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Application No. 60/214,878, filed June 28, 2000.

BACKGROUND OF THE INVENTION

10 The present invention relates to embedding data in a JPEG2000 file format.

At the core of the JPEG 2000 structure is a wavelet based compression methodology that provides for a number of benefits over the previous Discrete Cosine Transformation (DCT) compression methods used in the existing JPEG format.

Essentially, wavelets are mathematical expressions that encode the image in a continuous
15 stream; thereby avoiding the tendency toward visible artifacts that can sometimes result from DCT's division of an image into discrete compression blocks.

JPEG 2000 wavelet technology can provide as much as a 20% improvement in compression efficiency over existing JPEG DCT compression methods. JPEG 2000 wavelet technology also provides for both lossy and lossless compression, as opposed to
20 the lossy technique used in the original JPEG, which can lead to image degradation at high compression levels. In addition, because the JPEG 2000 format includes much richer content than existing JPEG files, the bottom line effect is the ability to deliver a Flashpix-

level of information in a compressed image file that is 20% smaller than baseline JPEG and roughly 40% smaller than an equivalent Flashpix file.

Another inherent benefit of JPEG 2000's use of wavelet technology is the ability to progressively access the encoded image in a smooth continuous fashion without having to download, decode, and/or print the entire file. In a way this allows for a virtual file system within the image file that can be flexibly arranged by the image providers to best suit the way that their users will need to access the information. For instance a "progressive-by-resolution" structure would allow the image information to stream to the user by starting with a low-resolution version and then progressively adding higher resolution as required. On the other hand, a "progressive-by-quality" structure might begin with a full resolution version but with minimal color data per pixel and then progressively add more bits per pixel as required.

Referring to FIG. 1, a conforming file for the JPEG2000 standard is typically described as a sequence of boxes, some of which contain other boxes. An actual file need not contain all of the boxes shown in FIG. 1, may contain different counts of the boxes, and/or could use the boxes in different positions in the file. A more complete description of the contents of these boxes is discussed in JPEG2000 Image Coding System: Compound Image File Format, JPEG2000 Part VI committee Draft, 9, March 2001.

Schematically, the hierarchical organization of boxes in a JPEG2000 file is shown in FIG.

2. Boxes with dashed borders are optional in conforming JPEG2000 files. However, an optional box may define mandatory boxes within that optional box. In this case, if the optional box exists, the mandatory boxes within the optional box normally exist. FIG. 2

illustrates only the containment relationship between the boxes in the file. A particular order of those boxes in the file is not generally implied. Referring to FIGS. 3A-3D, a list of exemplary boxes that may be used in a JPEG2000 file are illustrated.

A JPEG2000 file may contain metadata boxes with intellectual property right information or vendor specific information. In this manner the JPEG200 file may be annotated with intellectual property rights information. In particular, the metadata will normally provide the ability to include copyright information, such as the proper copyright ownership of image files. This helps alleviate long held concerns regarding the unauthorized appropriation of image files without the copyright owners consent. In this manner, at least the copyright information will be provided together with the JPEG2000 file and the image described therein.

A JPEG2000 file may also include a UUID (universal unique identifier) box that contains vendor specific information. There may be multiple UUID boxes within the file. The UUID box is intended to provide additional vendor specific information for particularized applications, which would normally reflect information regarding the rendering or usage of the image contained within the file. However, the content to be provided within the UUID box is undefined.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates JPEG2000 file elements and structure.

FIG. 2 illustrates conceptual structure of a JPEG2000 file.

FIGS. 3A-3D describe boxes used in a JPEG2000 file.

FIG. 4 illustrates a metadata box of a JPEG2000 file.

FIG. 5 illustrates a UUID box of a JPEG2000 file.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

5 What may be observed from the file format used for JPEG2000 is that nearly all the boxes contain data relevant to the rendering of the image itself, which is what would be expected from an image file format for a particular type of image, such as JPEG2000. Further, the image file format has been extended to include copyright information which is likewise of particular interest for the creator of the document. After
10 consideration of the JPEG2000 file format and the constructs provided by the JPEG2000 file format, the present inventors came to the startling realization that the previously recognized uses of the JPEG2000 metadata box may be extended, while not extending or otherwise modifying the file format in a non-compliant manner, to include data that is representative of a description of the content depicted by the JPEG2000 file or to otherwise
15 provide interactivity with the rendered image. The JPEG200 file format was intended to be a self-contained image description format for the rendering of an image and was not intended to support a description of the content of the image nor provide interactivity with the rendered image. Normally, if additional interactivity is desired for an image the file format is extended in a proprietary manner or otherwise an additional program is provided
20 which provides such a description of the content, such as a database, and interactivity with the rendered image, such as animation and game software. Preferably, the content of the metadata box does not change the visual appearance of the image.

Referring to FIG. 4, for example the metadata box may contain information regarding links to additional information, voice annotations, textual information describing the content of the image, hot spots, and object boundary information regarding objects within the image itself. Further, the textual information may relate to, for example, the title, the category, keywords, date of creation, time of creation, etc. In this manner, the textual information describes the content of the image to be rendered by a suitable JPEG2000 viewer but is typically free from changing the rendered image. In addition, this information is provided within the constructs of the JPEG2000 file format in a compliant manner so that all compliant JPEG2000 viewers will be able to render the image in a proper manner and in addition process the additional information, if desired. It is to be understood that the metadata box is preferably in XML format, however, any format may be used, if desired.

Referring to FIG. 5, after realizing the potential extension to the JPEG2000 file format the present inventors likewise determined that the UUID box may contain information regarding links to additional information, voice annotations, textual information describing the content of the image (e.g., actor, theme, genre, location, etc.), hot spots, and object boundary information regarding objects within the image itself. Further, the textual information may relate to, for example, the title, the category, keywords, date of creation, time of creation, etc. In this manner, the textual information describes the content of the image to be rendered by a suitable JPEG2000 viewer but is typically free from changing the rendered image. In addition, this information is provided within the constructs of the JPEG2000 file format in a compliant manner so that all

compliant JPEG2000 viewers will be able to render the image in a proper manner and in addition process the additional information, if desired. It is to be understood that the UUID box is preferably in XML format, however, any format may be used, as desired.

MPEG-7 is a description scheme that, at least in part, provides a description
5 of the content of video, such as actor, genre, etc. While MPEG-7 was specifically designed to relate to video content, the present inventors came to the realization that this video based scheme may be used for describing the content of an image file, namely JPEG2000 files, preferably in a compliant manner. Further, JPEG2000 specification does not define the syntax and semantics for the metadata that can be placed in the metadata and/or UUID
10 boxes in the file format. Therefore, a need exists for the specification of the syntax and semantics for the contents of these boxes, preferably in a standardized syntax and semantics specification that will permit the exchangeability of the metadata contents contained in these boxes. Referring to FIGS. 4 and 5, the present inventors came to the further realization that at least a portion of the MPEG-7 description schemes describing
15 video content may be suitable for use within the metadata boxes and/or UUID boxes of the JPEG2000 file format. This unlikely combination of file formats, namely JPEG2000 for image files and MPEG-7 describing video content, provides advantageous multi-standard interoperability. MPEG-7 is described in MPEG-7 Multimedia Description Schemes, Experimentation Model (XM) V 3.0, N3410, Geneva, May 2000; MPEG-7 Multimedia
20 Description Schemes, Working Draft (WD) V. 3.0, N3411, Geneva, May 2000; MPEG-7 Description Definition Language (DDL) WD 3.0, N3391, Geneva, May 2000; MPEG-7

Visual Part of XM 6.0, N3398, Geneva, May 2000; MPEG-7 Visual Part, Working Draft (WD) V. 3.0, N3399, Geneva, May 2000; all of which is incorporated by reference herein.

While the combination of MPEG-7 and JPEG2000 is a desirable goal, the
5 resulting file is preferably self-contained, in that all of the data necessary to render the
image is contained within the file format. In the same manner, preferably the metadata or
UUID information include the binary data necessary to execute or otherwise cause the
desired activity to be carried out. In contrast to the execution of binary code, MPEG-7 was
designed to provide a description of the content of the video media and accordingly lacked
10 suitable constructs for embedding binary data with the information. After the
determination of the need for embedding binary data within an MPEG-7 description
scheme, especially suitable for providing metadata or UUID data within a JPEG2000 file
format, the present inventors modified the previously existing MPEG-7 standard to include
a suitable technique for including binary data, which was not previously considered to have
15 any value.

A new description scheme was been developed, namely, "InlineMedia" that
permits the identification of the format of the media stream, such as for example, indicated
by a MediaFormat Description Scheme or a FileFormat (MIME-type) identifier. The audio
and/or visual material contained in an InlineMedia description may be either essence data
20 or audio and/or visual data representing other essence data, depending on its context. The
InlineMedia enables the description of audio and/or visual data located within the
description itself, without having to refer to a location external to the description.

The InlineMedia syntax may be as follows:

```
<!-- ##### -->
```

```
<!-- Definition of InlineMedia Datatype -->
```

```
5 <!-- ##### -->
```

```
<complexType name="InlineMediaType">
```

```
  <choice>
```

```
    <element name="MediaData16">
```

```
10      <simpleType>
```

```
        <restriction base="binary">
```

```
          <encoding value="hex">
```

```
        </restriction>
```

```
      </simpleType>
```

```
15    </element>
```

```
    <element name="MediaData64">
```

```
      <simpleType>
```

```
        <restriction base="binary">
```

```
          <encoding value="base64"/>
```

```
20        </restriction>
```

```
      </simpleType>
```

```
    </element>
```


</choice>

<attribute name="type" type="mpeg7:mimeType" use="required"/>

</complexType>

- 5 It is noted that <!-- is the start of a comment while --> is the end of a comment. Likewise choice provides a set of options, with the first option being binary data encoded in base 16 and the second option being binary data encoded in base 64. Other bases may likewise be used, as desired. The attribute name indicates the data type, such as MPEG data and their format, and whether this attribute is included in the description.

10

Summary of InlineMediaType

| | |
|-----------------|--|
| InlineMediaType | A descriptor for specifying media data embedded in the description. |
| MediaData16 | Specifies binary media data encoded as a textual string in base-16 format. |
| MediaData64 | Specifies binary media data encoded as a textual string in base-64 format. |
| Type | Specifies the MIME type of media data. |

15

20

InlineMedia Example

<myInlineMedia type="image/jpeg">

<MediaData16>98A34F10C5094538AB93873262522DA3</MediaData16>

</myInlineMedia>

The binary code embedded within the InlineMedia may be, for example, executable code, audio segments, video segments, and still images.

The InlineMedia descriptor is preferably included within the MediaLocation specification in MPEG-7, by modification of the MediaLocator specification.

The MediaFormat syntax may be as follows:

```
<!-- ##### -->
<!-- Definition the Media Format DS -->
10 <!-- ##### -->

<complexType name="MediaFormat">
    <element name="FileFormat" type="mds:ControlledTerm"/>
    <element name="System" type="mds:ControlledTerm" minOccurs="0"/>
    15 <element name="Medium" type="mds:ControlledTerm" minOccurs="0"/>
    <element name="Color" type="mds:ControlledTerm" minOccurs="0"/>
    <element name="Sound" type="mds:ControlledTerm" minOccurs="0"/>
    <element name="FileSize" type="nonNegativeInteger" minOccurs="0"/>
    <element name="Length" type="mds:TimePoint" minOccurs="0"/>
    20 <element name="AudioChannels" type="nonNegativeInteger" minOccurs="0"/>
    <element name="AudioLanguage" type="language" minOccurs="0"/>
    <element name="id" type="ID"/>
</complexType>
```

Summary of MediaFormat

| | | |
|----|---------------|--|
| | MediaFormat | Description of the storage format of the media. |
| | id | Identification of the instance of the media format description. |
| 5 | FileFormat | The file format or MIME type of the audio and/or video content instance. |
| | System | The video system of the audio and/or video content (e.g., PAL, NTSC). |
| 10 | Medium | The video system of the audio and/or video content is sotred (e.g., tape, CD, DVD). |
| | Color | The color domain of the audio and/or video content (e.g., color, black/white, colored). |
| | Sound | The sound domain of the audio and/or video content (e.g., no sound, stereo, mono, dual, surround, 5.1, dolby digital). |
| 15 | FileSize | The size, in byte for example, of the file where the audio and/or video content is stored. |
| | Length | The duration of the audio and/or video content. |
| | AudioChannels | The number of audio channels in the audio and/or video content. |
| 20 | AudioLanguage | The language used in the audio of the audio and/or video content. |

Also, the previously existing MediaLocator of MPEG-7 is extended by adding the InlineMedia as follows:

```
<complexType name="MediaLocator">
```

```
    <choice>
```

```
5         <sequence>
```

```
            <element name="MediaURL" type="mds:MediaURL"/>
```

```
            <element name="MediaTime" type="mds:MediaTime"
```

```
minOccurs="0"/>
```

```
        </sequence>
```

```
10        <element name="MediaTime" type="mds:MediaTime"/>
```

```
        <element name="InlineMedia" type="mds:InlineMedia"/>
```

```
    </choice>
```

```
</complexType>
```

```
15                               MediaLocator Example
```

```
<MediaLocator>
```

```
    <InlineMedia>
```

```
        <FileFormat>mp3</FileFormat>
```

```
        <MediaData>98A34F12348942323423AB2342</MediaData>
```

```
20    </InlineMedia>
```

```
</MediaLocator>
```

An alternative implementation assumes that the media data can be placed at an arbitrary location in the JPEG2000 files. In this case a byte offset may be used to locate the binary data. In this case, the MediaLocator is alternatively modified as follows:

```

5  <complexType name="MediaLocator">
    <choice>
        <sequence>
            <element name="MediaURL" type="mds:MediaURL"/>
            <element name="MediaTime" type="mds:MediaTime"
10  minOccurs="0"/>
        </sequence>
        <sequence>
            <element name="MediaURL" type="mds:MediaURL"/>
            <element name="ByteOffset" type="nonNegativeInteger"
15  minOccurs="0"/>
        </sequence>
        <element name="MediaTime" type="mds:MediaTime"/>
    </choice>
</complexType>
20

```

In this embodiment of MediaLocator, the MediaURL points to the JPEG2000 file itself. The format of the media is specified by the MediaFormat. The ByteOffset may be an

absolute offset within the file, a relative offset, or otherwise indicating a location within the file.

Another embodiment of the present invention includes another class of applications, namely, a bounding region of a portion of the image and associating metadata and information with this bounding region(s). The information is typically related to the objects (or image regions) that are defined by the bounding region. The metadata box and/or the UUID box in the JPEG200 file format may be utilized to store descriptors and data that define and identify the bounding regions as well as data associated with the regions, such as object specific URL links, voice annotation, and textual annotation. One of many applications of such data is user interaction with images where the users interactively discover and consume information that relate to the content of the image.

While any suitable syntax may be used to define the bounding region, the bounding region is preferably expressed in XML. Further, the XML is preferably expressed in the form defined by MPEG-7 so that the JPEG2000 file and the MPEG-7 portion are compliant with the respective standards.

Within the MPEG-7 standard the bounding region may be achieved by using the Still Region Description Scheme. The Still Region Description Scheme is derived from the Segment Description Scheme. The Segment Description Scheme is used to specify the structure of spatial and temporal segments of visual data such as images and video in general. Segments can be decomposed into other segments. The Still Region Description Scheme is used to specify a spatial type of segment in still images or a single video frames.

The Segment Description Scheme and the Still Region Description Scheme

may be as follows:

```
<!-- ##### -->
5  <!-- Definition of "Segment DS" -->
<!-- ##### -->

<!-- Definition of datatype of the decomposition -->
<simpleType name="DecompositionDataType" base="string">
10  <enumeration value="spatial"/>
    <enumeration value="temporal"/>
    <enumeration value="spatio-temporal"/>
    <enumeration value="MediaSource"/>
</simpleType>
15
<!-- Definition of the decomposition -->
<complexType name="SegmentDecomposition">
    <element ref="Segment" minOccurs="1" maxOccurs="unbounded"/>
    <attribute name="DecompositionType" type="mds:DecompositionDataType"
20  use="required"/>
    <attribute name="Overlap" type="boolean" use="default" value="false"/>
    <attribute name="Gap" type="boolean" use="default" value="false"/>
</complexType>
```

<element name="Segment" type="mds:Segment"/>

<!-- Definition of the Segment itself -->

<complexType name="Segment" abstract="true">

5 <element name="MediaInformation" type="mds:MediaInformation"
 minOccurs="0" maxOccurs="1"/>

 <element name="CreationMetaInformation" type="mds:CreationMetaInformation"
 minOccurs="0" maxOccurs="1"/>

 <element name="UsageMetaInformation" type="mds:UsageMetaInformation"
10 minOccurs="0" maxOccurs="1"/>

 <element name="StructuredAnnotation" type="mds:StructuredAnnotation"
 minOccurs="0" maxOccurs="unbounded"/>

 <element name="MatchingHint" type="mds:MatchingHint" minOccurs="0"
 maxOccurs="unbounded"/>

15 <element name="PointOfView" type="mds:PointOfView" minOccurs="0"
 maxOccurs="unbounded"/>

 <element name="SegmentDecomposition" type="mds:SegmentDecomposition"
 minOccurs="0" maxOccurs="unbounded"/>

 <attribute name="id" type="ID" use="required"/>

20 <attribute name="href" type="uriReference" use="optimal"/>

 <attribute name="idref" type="IDREF" refType="Segment" use="optional"/>

</complexType>

Summary of SegmentDecomposition

SegmentDecomposition

Decomposition of a segment into one or more segments.

5 DecompositionDataType

Datatype defining the kind of segment decomposition. The possible kinds of segment decomposition are spatial, temporal, spatio-temporal, and media source. The bounding regions may be, for example, spatial segments.

10 DecompositionType

Attribute, which specifies the decomposition type of a segment.

Overlap Boolean, which specifies if the segments resulting from a segment decomposition overlap in time or space. The bounding regions in the image may overlap.

15 Gap Boolean, which specifies if the segments resulting from a segment decomposition leave gaps in time or space.

Segment Set of segments that form the composition.

Summary of Segment

20 Segment Abstract structure which represents a fragment or section of the audio and/or video content. For example, a segment may be a region in an image or a moving region in a video sequence. A segment can be decomposed into other segments through the

SegmentDecomposition. This may be used to specify the object's shape, if needed, within a bounding region, where the outline of the object is specified in terms of a decomposition of the bounding region.

5 id Identifier of a video segment. This may be used to uniquely identify multiple bounding regions, spatial segments, in an image.

DecompositionDataType

10 Datatype defining the kind of segment decomposition. The possible kinds of segment decomposition are spatial, temporal, spatio-temporal, and media source.

MediaInformation

Media information relates to the segment and its descendants.

CreationMetaInformation

15 Creation Meta Information relates to the segment and its descendants. This may be used to associate data with segments, such as URL, audio files, etc.

UsageMetaInformation

Usage Meta Information relates to the segment and its descendants.

SegmentDecomposition

20 Decomposition of the segment into sub-segments.

Annotation Textual annotation and description of people, animals, objects, actions, places, time, and/or purpose which are instantiated in the

segment. This may be used to associate textual annotations with the bounding regions.

```

5  <!-- ##### -->

    <!-- Definition of "StillRegion DS" -->

    <!-- ##### -->

    <element name="StillRegion" type="mds:StillRegion" equivClass="Segment"/>
10 <complexType name="StillRegion" base="mds:Segment" derivedBy="extension">
    <element ref="ColorSpace" minOccurs="0" maxOccurs="1"/>
    <element ref="ColorQuantization" minOccurs="0" maxOccurs="1"/>
    <element ref="DominantColor" minOccurs="0" maxOccurs="1"/>
    <element ref="ColorHistogram" minOccurs="0" maxOccurs="1"/>
15 <element ref="BoundingBox" minOccurs="0" maxOccurs="1"/>
    <element ref="RegionShape" minOccurs="0" maxOccurs="1"/>
    <element ref="ContourShape" minOccurs="0" maxOccurs="1"/>
    <element ref="ColorStructureHistogram" minOccurs="0" maxOccurs="1"/>
    <element ref="ColorLayout" minOccurs="0" maxOccurs="1"/>
20 <element ref="CompactColor" minOccurs="0" maxOccurs="1"/>
    <element ref="HomogeneousTexture" minOccurs="0" maxOccurs="1"/>
    <element ref="TextureBrowsing" minOccurs="0" maxOccurs="1"/>
    <element ref="EdgeHistogram" minOccurs="0" maxOccurs="1"/>

```

```

<element ref="SpatialConnectivity" type="boolean" use="required"/>

<!-- Restriction of refType to StillRegion DS -->

<attribute name="idref" type="IDREF" refType="StillRegion" use="optional"/>

</complexType>

```

5

StillRegion Summary

| | | |
|----|---------------------|---|
| 10 | StillRegion | <p>Set of pixels from an image or a frame in a video sequence.</p> <p>It is noted that no motion information should be used to describe a still region. Still image can be natural image or synthetic images. A still image is a particular case of a still region. The pixels do not need to be connected (see the SpatialConnectivity attribute).</p> |
| 15 | SpatialConnectivity | <p>Boolean which specifies if a still region is connected in space, i.e. connected pixels.</p> |
| | ColorSpace | <p>Description of the color space used for the color of the still region.</p> |
| | ColorQuantization | <p>Description of the color quantization used for the color of the still region.</p> |
| 20 | DominantColor | <p>Description of the dominant color of the still region.</p> |
| | ColorHistogram | <p>Description of the color histogram of the region. This may be used to embed a low-level color description to bounding regions, when desired.</p> |

| | |
|-------------|---|
| BoundingBox | Description of a bounding region containing the region. This is used to describe the bounding region as a region, such as a rectangular region. |
|-------------|---|

5 Using the aforementioned specification the bounding region in a JPEG2000 image may be described as spatial segments and the descriptor BoundingBox may be used to define the locations and dimensions of bounding region(s), and each region is identified by an id, which is preferably unique.

10 Embedding of textual information, such as annotations, may be implemented by the structured annotation description scheme. Each segment can reference the structured annotation description scheme individually and at multiplicities identified by their corresponding identifiers. The StructuredAnnotation Description Scheme may be as follows:

15 <!-- ##### -->
 <! -- Definition of StructuredAnnotation DS -->
 <! -- ##### -->

 <element name="TextAnnotation" type="mds:TextualDescription"/>
 20 <element name="structuredAnnotation" type="mds:StructuredAnnotation"/>
 <complexType name="StructuredAnnotation" type="mds:StructuredAnnotation"/>
 <complexType name="StructuredAnnotation">
 <element name="Who" type="mds:ControlledTerm" minOccurs="0"/>

| | | |
|----|----------------------|--|
| | | <element name="WhatObject" type="mds:ControlledTerm" minOccurs="0"/> |
| | | <element name="WhatAction" type="mds:ControlledTerm" minOccurs="0"/> |
| | | <element name="Where" type="mds:ControlledTerm" minOccurs="0"/> |
| | | <element name="When" type="mds:ControlledTerm" minOccurs="0"/> |
| 5 | | <element name="Why" type="mds:ControlledTerm" minOccurs="0"/> |
| | | <element name="TextAnnotation" type="mds:TextualDescription" |
| | | minOccurs="0"/> |
| | | <attribute name="id" type="ID"/> |
| | | <attribute ref="xml:lang"/> |
| 10 | | </complexType> |
| | | StructuredAnnotation Summary |
| | TextAnnotation | Free textual annotation. |
| | StructuredAnnotation | Textual free annotation and description of people, animals, |
| 15 | | objects, actions, places, time, and/or purpose. |
| | Who | Textual description of people and animals. May be from a |
| | | thesaurus or a controlled vocabulary. |
| | WhatObject | Textual description of objects. May be from a thesaurus or a |
| | | controlled vocabulary. |
| 20 | WhatAction | Textual description of actions. May be from a thesaurus or a |
| | | controlled vocabulary. |
| | Where | Textual description of places. May be from a thesaurus or a |
| | | controlled vocabulary. |

| | | |
|---|------------|---|
| | When | Textual description of time. May be from a thesaurus or a controlled vocabulary. |
| | Why | Textual description of purpose May be from a thesaurus or a controlled vocabulary. |
| 5 | Annotation | Textual free annotation and description of people, animals, objects, actions, places, time, and/or purpose. |
| | id | Identifier for an instantiation of the StructuredAnnotation Description Scheme. |

10 Embedding of universal resource locators (URL's) (identifier for information outside of the JPEG2000 file) for each bounding region may be realized using the RelatedMaterial description. The RelatedMaterial description scheme is referenced by the CreationMetaInformation DS. Each segment (e.g., each bounding region) references CreationMetaInformation DS, multiple times, if desired. The RelatedMaterial DS may be

15 specified as follows:

```

<!-- ##### -->

<!-- Definition the RelatedMaterial DS -->

20 <!-- ##### -->

<DSType name="RelatedMaterial">

    <attribute name="id" datatype="ID"/>

```

```

<attribute name="Master" datatype="boolean" default="true" required="false"/>

<DTypeRef name="MediaType" type="controlledTerm"/>

<DTypeRef type="MediaLocator" minOccurs="0"/>

<DTypeRef type="MediaInformation" minOccurs="0"/>

5    <DTypeRef type="CreationMetaInformation" minOccurs="0"/>

    <DTypeRef type="UsageMetaInformation" minOccurs="0"/>

</DType>

```

RelatedMaterial Summary

| | | |
|----|-------------------------|--|
| 10 | RelatedMaterial | Description of the materials containing additional information about the audio and/or video content. |
| | Master | Boolean attribute that allows to identify if the referenced related material is the master. |
| | MediaType | The media type of the referenced related material (e.g., web page, audiovisual media, a printed book). |
| 15 | MediaLocator | The locator of the referenced related material. |
| | MediaInformation | The media information description of the referenced related material. |
| | CreationMetaInformation | |
| 20 | | The creation meta information description of the referenced related material. |
| | UsageMetaInformation | |

The usage meta information description of the referenced related material.

In another embodiment the media data may be included in the UUID box in the JPEG2000 file. In this embodiment the MPEG-7 description schemes are suitable for use in their previously existing format. Typically the UUID box is implicitly referenced from the metadata box via the MediaFormat Description Scheme. The MediaProfile DS and the MediaInformation DS may be as follows:

```
10 <!-- ##### -->
    <!-- Definition the MediaProfile DS -->
    <!-- ##### -->

15 <DSType name="MediaProfile">
    <attribute name="id" datatype="ID"/>
    <DSTypeRef type="MediaInformation"/>
    <DSTypeRef type="MediaFormat"/>
    <DSTypeRef type="MediaCoding" minOccurs="0" maxOccurs="*" />
20 <DSTypeRef type="MediaInstance" minOccurs="0" maxOccurs="*" />
    </DSType>
```

Summary of MediaProfile

| | |
|---------------------|---|
| MediaProfile | DS describing one profile of the media being described. |
| id | Identification of the instance of the MediaProfile description. |
| MediaIdentification | Identification of the master media profile. |
| MediaFormat | Description of the storage format of the master media profile. |
| 5 MediaCoding | Description of the coding parameters of the master media profile. |
| MediaInstance | Description and the localization of the master media profile. |

```

10  <!-- ##### -->
    <!-- Definition the MediaInformation DS -->
    <!-- ##### -->

    <DSType name="MediaInformation">
15      <attribute name="id" datatype="ID"/>
      <DSTypeRef type="MediaProfile" maxOccurs="*" />
    </DSType>

```

Summary of MediaInformation

20

MediaInformation The MediaInformation DS contains one or more MediaProfile DSs. Each MediaInformation DS is related to one reality. For example, a concert may have been recorded

in audio and in audio-visual media. Afterwards each media may be available in different format, e.g., the audio media in CD, and the audio-visual media in MPEG-1, MPEG-2, and MPEG-4. This will imply four MediaProfiles for the same reality.

id Identification of the instance of the MediaProfile description.

MediaProfile DS describing one profile of the essence being described.

In this embodiment, when the MediaLocator within the Related Media description points at the JPEG2000 file itself via MediaURL, the client application implicitly knows that the related media is contained in a UUID box within this same file containing the XML box. The UUID is referenced through Media Format description. The application will then locate the UUID box with the matching ID in the file and read its contents. The format of the audio media (e.g., mp3) that is contained in the UUID box may be specified a priori by the owner of the UUID format. The mechanism for referring to the JPEG2000 file itself and the UUID from the XML box is summarized below, suching the existing MPEG-7 description schemes and their hierarchical structure:

20 RelatedMaterial

MediaType

Audio

MediaLocator

URL:JPEG2000 file

MediaInformation

MediaProfile

MediaFormat

5 UUID

10 The XML box is equipped by a mechanism to refer to the UUID box that contains the data, as described above. A format needs to be specified for the UUID box in order to organize the data within and associate the data with different regions and different media types. This format is typically vendor specific and identified by the UUID.

15 The following format for the UUID box is one potential example. It assumes that all the embedded data is stored in one single UUID box, provided that the data are within the same file. Data associated with different regions are identified according to their corresponding region ID. Types of data are also specified. The Region Data Length is included to minimize parsing during navigation amongst different regions as the user interacts with the image. The media Data Length is included to enable rapid navigation of data embedded within the same region.

20

UUID Box Format

Comment

| | | |
|----|--------------------|---|
| | ID | The ID of the particular UUID box is specified by the MediaInformation/MediaFormat description referenced in the RelatedMaterial description in the XML box. |
| 5 | Region ID | Matches the ID of the Still Region described by the StillRegion description in the XML box. |
| | Region Data Length | Total length of data associated with this region. |
| | Media Type | Media Type corresponds to the value of the MediaType descriptor in RelatedMaterial description in the XML box (it may be mapped to a binary code in the UUID box) |
| 10 | Media Data Length | |
| | Media Data | |
| | ... | |
| | Media Type | |
| | Media Data Length | |
| 15 | Media Data | |
| | ... | |
| | Region ID | |
| | Region Data Length | |
| | Media Type | |
| 20 | Media Data Length | |
| | Media Data | |
| | ... | |

It may be noted that the Region ID in the above table may be generalized to an “Object ID”. The Object ID may then refer to any XML object, i.e., any description that is identified by an ID. In that case, a Person Description may have an audio annotation associated with it, or a Summary Description may have executable software associated with it. MPEG-7 does support identification of XML descriptions using unique identifiers.

Summary of MPEG-7 tools used in the UUID box of JPEG200

| | Embedded Information | MPEG-7 Tool | JPEG2000 File Format |
|----|-----------------------------|---------------------|---|
| | | | Structure |
| 10 | Bounding Region(s) | Still Region DS | XML Box |
| | Textual Annotation | Annotation DS | XML Box |
| | URL Link | Related Material DS | XML Box |
| | Audio/Voice Annotation Data | Related Material DS | XML Box: indicates Media Type as “Audio” and contains reference to the UUID Box; contains the audio data. |
| 15 | | | |
| | Executable Code | Related Material DS | XML Box: indicates Media Type as “executable” and contains reference to the ID of the UUID box containing the executable; UUID Box: contains the executable code. |
| 20 | | | |

In a multi-level implementation of the system, a server may first provide the client the image data, the bounding regions, and the type and format of the data associated with the bounding regions. The data that is of further interest to the user may then be delivered upon user's request.

- 5 If desired, MPEG-7 compliant data/information may be considered the MPEG-7 specification as it exists (or substantially similar) to the date of filing of this application.

All the references cited herein are incorporated by reference.

- 10 The terms and expressions that have been employed in the foregoing specification are used as terms of description and not of limitation, and there is no intention, in the use of such terms and expressions, of excluding equivalents of the features shown and described or portions thereof, it being recognized that the scope of the invention is defined and limited only by the claims that follow.